

ПОДХОДЫ К ОБУЧЕНИЮ РАСПРЕДЕЛЕННОГО КОЛЛЕКТИВА РОБОТОВ

Ворманов И.А.*, Банников И.К., Евсегнеев О.А.

Уральский федеральный университет имени первого Президента России Б.Н. Ельцина, г. Екатеринбург, Россия

*E-mail: via96rus@gmail.com

APPROACHES TO TEACHING A DISTRIBUTED ROBOT COLLECTIVE

Vormanov I.A.*, Bannikov I.K., Evsegneeov O.A.

Ural Federal University, Yekaterinburg, Russia

Annotation. The scope of distributed intelligence is currently very large. The basis of the effective functioning of this intellect is the process of its learning. To improve the quality of education, it is composed of two parts: the training of each individual individually and the training of the team as a whole.

Область применения распределенного интеллекта в настоящее время очень велика. Основу эффективного функционирования данного интеллекта составляет процесс его обучения. Для повышения качества обучения оно составляется из двух частей: обучение каждого индивида в отдельности и обучения коллектива в целом[1].

Для модификации поведения особи коллектива предлагается использовать следующий набор модификаторов:

- инициативность - показатель состоятельности оценки найденного направления в качестве направления развития коллектива;
- эффективность принятия решения - показатель качества принятия решений, основанный на объеме наработанных шаблонов поведения, а также их комбинаций;

Модификация взаимодействия особей коллектива влияет на следующие параметры:

- коэффициент лидерства - показатель влияния особи на принятие решения другими индивидами коллектива;
- коэффициент допустимой нагрузки - оценка возможности распределения вычислительной нагрузки между особями коллектива;
- оценка возможности оказания помощи другой особи коллектива в принятии решения для повышения его эффективности.

Серьезной проблемой в функционировании распределенного интеллекта является эффективное разбиение поставленной задачи на подзадачи и распределение их между особями коллектива[2].

Данный подход позволяет частично устранить данный недостаток, путем расширения набора варьируемых параметров, что позволяет достичь лучшего согласования особей коллектива между собой.

1. Ворманов И.А., Банников И.К., Евсегнеев О.А., Пепелев А.М. Обучение распределенного коллектива роботов // V Международная молодежная научная конференция, посвященная памяти Почетного профессора УрФУ В.С. Кортova Физика. Технологии. Инновации. с.50 (2018)
2. Карпов В. Э. Коллективное поведение роботов. Желанное и действительное // Современная мехатроника. Сб. научн. Трудов Всероссийской научной школы, Орехово-Зуево (2011)

РАЗВИТИЕ МОДЕЛИ МЕНЕДЖЕРА ЗАДАЧ С ПРИОРИТЕТАМИ В C++

Язев М.К.*, Кузнецов М.А., Неудачин И.Г.

Уральский федеральный университет имени первого Президента России
Б.Н. Ельцина, г. Екатеринбург, Россия

*E-mail: maksyazev@mail.ru

THE DEVELOPMENT OF THE TASK MANAGER MODEL WITH PRIORITIES IN C++

Yazev M.K.*, Kuznetsov M.A., Neudachin I.G.

Ural Federal University, Yekaterinburg, Russia

At the moment, the C++ programming language is rather poorly optimized approach to task management, and there is no task Manager. We create a new task Manager model with priorities based on the existing "futures and promises" approach. The C++ standard already has some basic task management tools, such as: `std::future`, `std::promise`, `std::packaged_task`, `std::async`. The `std::priority_queue` class is a priority queue base that should be improved.

В данный момент в языке программирования C++ достаточно слабо оптимизирован подход к управлению задачами, а также нет менеджера задач. Предполагается создать новую модель менеджера задач с приоритетами на основе существующего подхода "futures and promises". В стандарте C++[2] уже существуют некоторые базовые части в классах `std::future`, `std::promise`, `std::packaged_task`, `std::async` – по управлению задачами, `std::priority_queue` – по очередям с приоритетами, которые станут базой для дальнейшего улучшения.

В новой библиотеке модернизируется концепция "futures and promises"[1] к многофункциональности задач, возможности их одновременного выполнения, качественного распределения машинных ресурсов на самые приоритетные задачи в каждый момент времени, решается проблема автоматического запуска задач в многопоточной среде.

Для разработки данной библиотеки на языке C++ за основу взят фреймворк Qt, совмещающий множество необходимых современных разработок, в частности механизм сигналов и слотов для кооперации управления задачами в реальном времени.